Scalable Data Management Using Metadata and Provenance

Ethan Miller[*] • Darrell Long[*] • Margo Seltzer[†]
[*]UC Santa Cruz and [†]Harvard University

Motivation

- HEC file systems are growing
  - Store petabytes (soon exabytes?) of data
  - Contain 10^9– 10^{10} files, if not more
- Hierarchical name spaces don't scale sufficiently
  - Namespace must become both broad and tall
  - Names are "static": difficult to assign multiple names to a single file
- Hierarchical name spaces are becoming difficult to use
  - File names often encode file properties
  - Users are starting to ignore the "real" file name
    - Obtain it via DB search and copy it to application

Project vision

- Goal: combine metadata gathering and indexing with search to build an extensible name space
  - Files are indexed automatically (and quickly!)
    - The index is the only file-tracking structure in the system
  - Files can be linked to one another
    - Links may be automatically generated or user-created
  - All file names and directories are search queries
- Benefits
  - Flexible file system naming
    - Find files based on names and characteristics
    - No disconnect between naming and search
    - Personalized name spaces can be created
  - Incorporation of provenance with other metadata: simplifies file management
  - Scalable name space: handles billions of files
  - Simpler & more efficient: one FS structure, not parallel structures

Challenges

Challenges

- Similar approaches are already coming for
  - The Web: Google
  - Desktops: Spotlight

Challenges

- Similar approaches are already coming for
  - The Web: Google
  - Desktops: Spotlight
- Existing solutions have issues

Challenges

- Similar approaches are already coming for
  - The Web: Google
  - Desktops: Spotlight
- Existing solutions have issues
  - Google uses too much hardware!

4

Challenges

- Similar approaches are already coming for
  - The Web: Google
  - Desktops: Spotlight
- Existing solutions have issues
  - Google uses too much hardware!
  - Spotlight is too slow and doesn't scale well

Challenges

- Similar approaches are already coming for
  - The Web: Google
  - Desktops: Spotlight
- Existing solutions have issues
  - Google uses too much hardware!
  - Spotlight is too slow and doesn't scale well
  - Neither includes provenance or locality-based search

Challenges

- Similar approaches are already coming for
  - The Web: Google
  - Desktops: Spotlight
- Existing solutions have issues
  - Google uses too much hardware!
  - Spotlight is too slow and doesn't scale well
  - Neither includes provenance or locality-based search
- Challenging because

Challenges

- Similar approaches are already coming for
  - The Web: Google
  - Desktops: Spotlight
- Existing solutions have issues
  - Google uses too much hardware!
  - Spotlight is too slow and doesn't scale well
  - Neither includes provenance or locality-based search
- Challenging because
  - Existing approaches don't scale well
    - Amount of metadata
    - Performance & reliability
    - Namespace size

4

Challenges

- Similar approaches are already coming for
  - The Web: Google
  - Desktops: Spotlight
- Existing solutions have issues
  - Google uses too much hardware!
  - Spotlight is too slow and doesn't scale well
  - Neither includes provenance or locality-based search
- Challenging because
  - Existing approaches don't scale well
    - Amount of metadata
    - Performance & reliability
    - Namespace size
  - File names are firmly entrenched: need backwards compatibility

Research thrusts

- Three main research thrusts in this project
- Gathering and managing metadata
  - File properties
  - File content
  - Provenance
- Metadata index construction and management
- File naming & search

Gathering and managing metadata

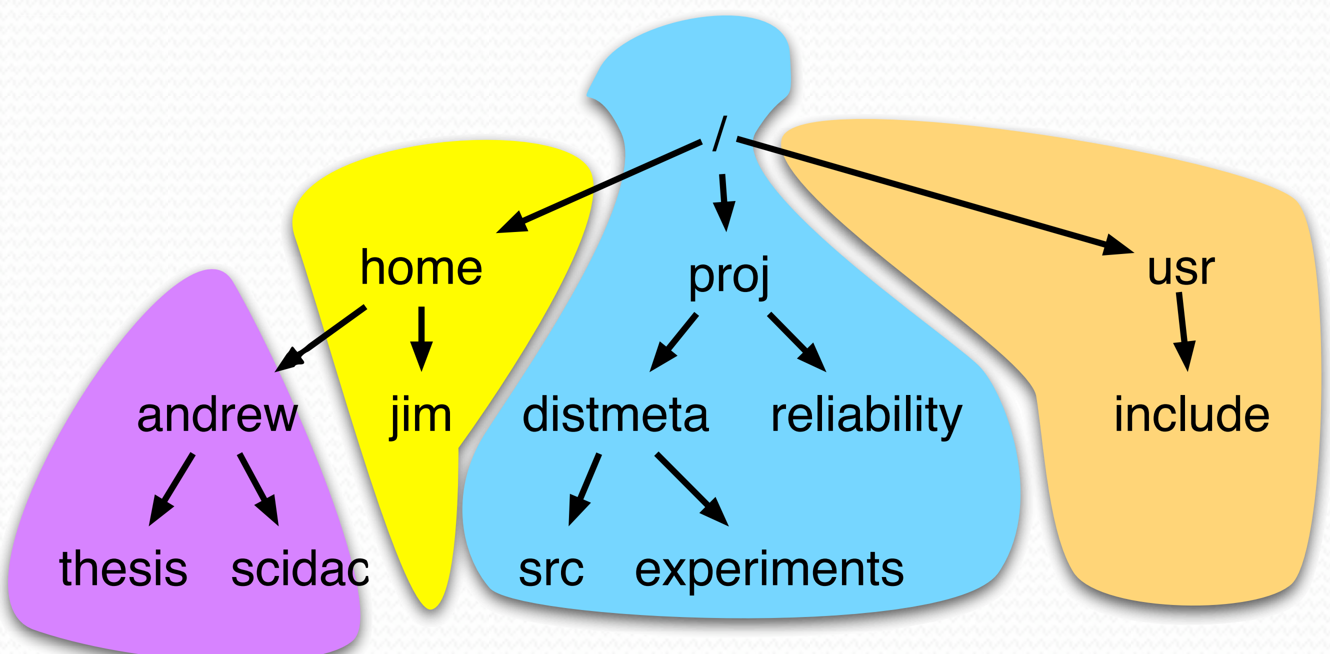
- Need to efficiently gather metadata
  - Straightforward for single-node files
  - More difficult to index content of a multi-gigabyte file
- Gathering provenance across a large-scale HEC system can also be difficult
- Need to provide an efficient, easy-to-use API for users and applications to provide metadata
  - Content-based
  - User-defined tags
  - Links between files

Indexing metadata

- HEC file systems have billions (10^9– 10^{10}) of files
  - Each file can require 1–100KB (or more) of metadata
  - Total index size is many terabytes
- Performance is critical
  - Search-based naming will be the *only* name lookup scheme
  - Existing database-style solutions aren't fast enough
- Our goal: leverage file system characteristics to build a fast, efficient index
  - Attribute locality
  - Search locality

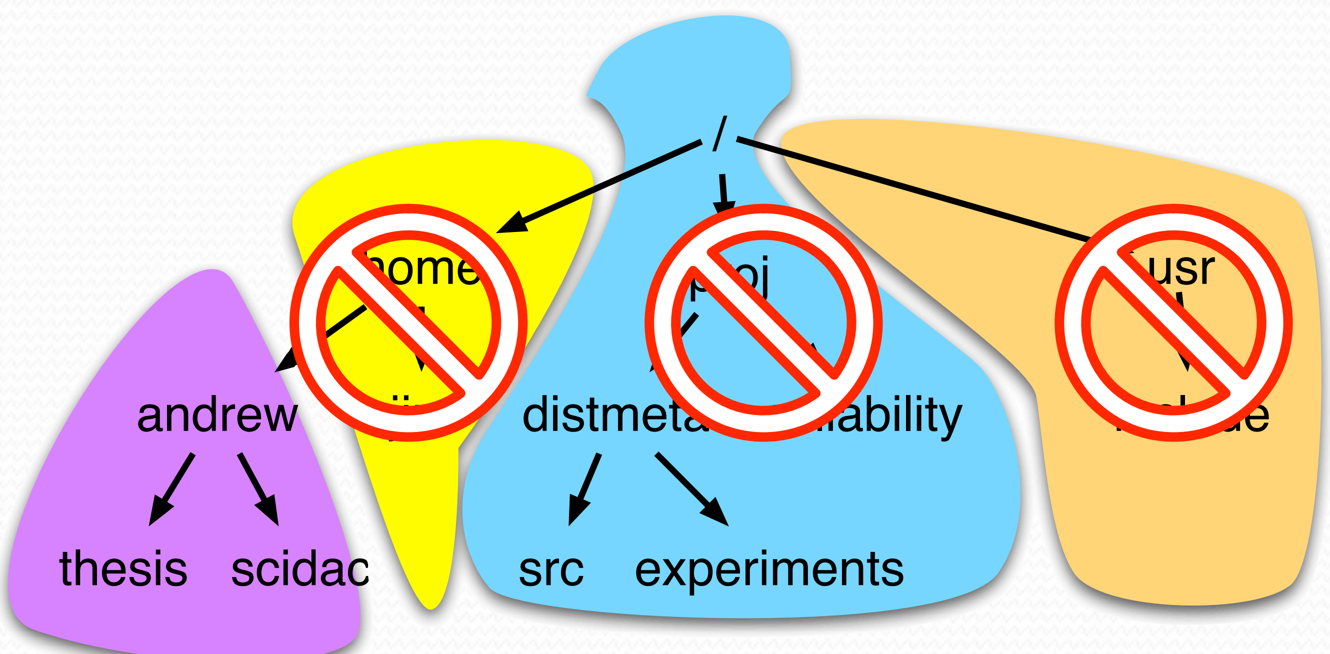
Spyglass: namespace partitioning

- Partition the index using the namespace
- Parts of the namespace are indexed in separate partitions
  - Exploits spatial locality
  - Allows index control at the granularity of sub-trees
  - Uses a simple greedy algorithm
- Partitions are stored sequentially on disk
  - Fast access to each partition



Spyglass: namespace partitioning

- Partition the index using the namespace
- Parts of the namespace are indexed in separate partitions
  - Exploits spatial locality
  - Allows index control at the granularity of sub-trees
  - Uses a simple greedy algorithm
- Partitions are stored sequentially on disk
  - Fast access to each partition



Spyglass: partition design

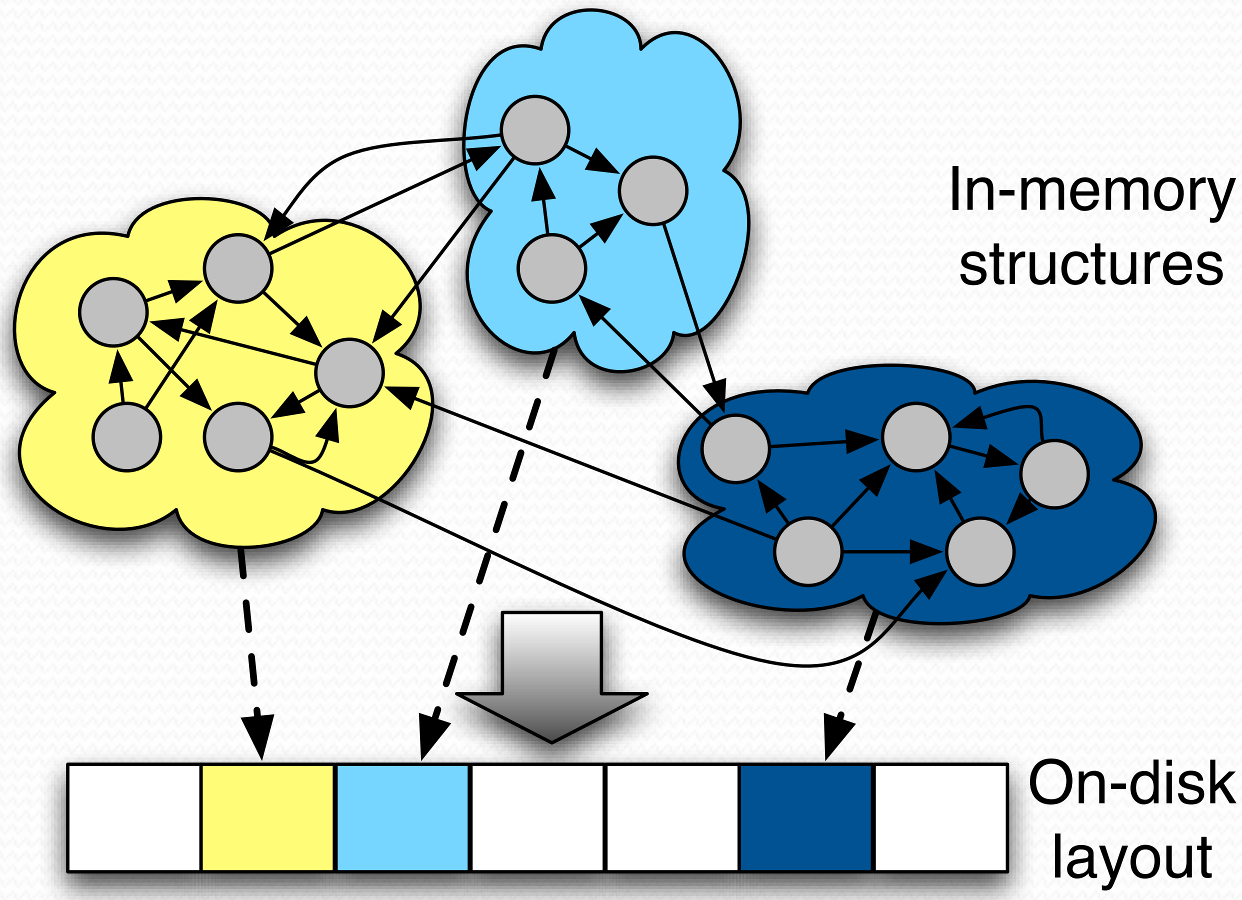
- Each partition stores metadata in a KD-tree
- Not explicitly tied to a particular index structure

- KD-trees
  - A multi-dimensional binary tree
  - Provides fast, multi-dimensional search
  - Allows a single index structure to be used

- Performance is bound by reading partitions from disk
- Partitions are managed by a caching sub-system
  - Uses LRU
  - Captures the *likely* Zipf-like query distributions
  - Ensures popular partitions are in-memory

Ongoing indexing work

- Expand cluster-based indexing to non-hierarchical file systems
  - Develop good approaches for clustering into sub-indexes
  - Ensure that clustered indexing scales: leverage locality
- Explore techniques for fast updating of indexes
  - Use index structures that can be efficiently updated?
  - Periodically rebuild indexes?
- Improve efficiency
  - Use Bloom filters (summaries) to query fewer clusters
  - Prune metadata and provenance to limit index size
  - Use NVRAM (flash, phase-change) for indexes
- Improve reliability



Searching and naming

- Replace hierarchical naming with dynamic names
  - Names are queries into the metadata index
  - Queries have to be fast!
- Challenges
  - Simple names for simple queries
  - Expressive, flexible language for complex queries
    - Allow searches across metadata, provenance, and file relationships
  - Language must allow for extensible names
    - File system can't dictate metadata schema
    - Different users will want different names
- Extend early work (PQL, QUASAR) to develop a usable naming scheme

11

Project plans

- Develop integrated index in a metadata server for an existing file system (*e.g.*, PVFS, Ceph)
  - MDS can be tested on its own in the lab
  - MDS can be moved to the field (after sufficient hardening)
- Explore approaches to scalability & partitioning
  - Algorithms for scalable metadata gathering and indexing
  - Factors that impact scalability & partitioning effectiveness
  - Use of non-volatile memory to improve performance
- Develop methods to use provenance to guide workflow
- Investigate techniques for per-user customization of the name space and index
  - Provenance & security
  - Per-user usage history and preferences
- Refine a path query language

12

Questions?

UC Santa Cruz
Storage Systems Research Center

Ethan Miller

Darrell Long

Harvard
SYRAH lab

Margo Seltzer

13

Questions?

- These are project *goals*
  - We've done some early work on these areas
  - Much remains to be done

UC Santa Cruz
Storage Systems Research Center

Ethan Miller

Darrell Long

Harvard
SYRAH lab

Margo Seltzer

Questions?

- These are project *goals*
  - We've done some early work on these areas
  - Much remains to be done
- Feedback is welcome!

UC Santa Cruz
Storage Systems Research Center

Ethan Miller

Darrell Long

Harvard
SYRAH lab

Margo Seltzer

13